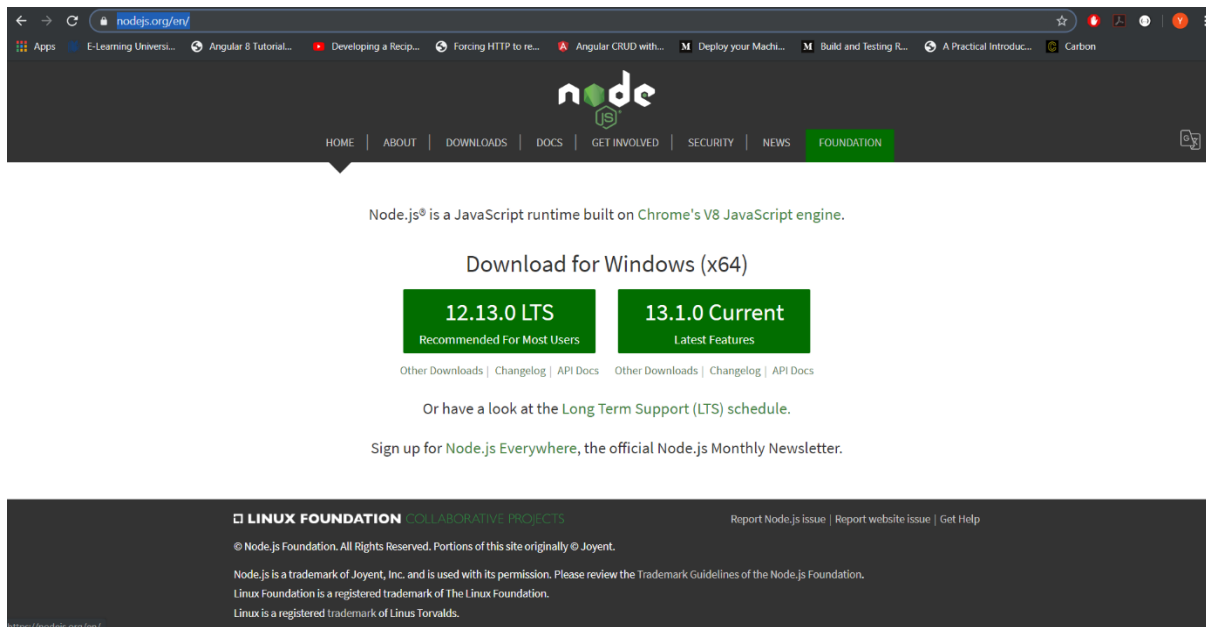


Angular Simple Tutorial

Sebelum kita koding, mari kita download dan install Angular dan Node JS.

Silahkan download Node JS terlebih dahulu di <https://nodejs.org/en/>.



Silahkan download versi Node JS yang paling **stable (Recommended for Most User)**.

Setelah di download, install-lah seperti biasa (install default).

Setelah selesai proses instalasi, bukalah *terminal* atau CMD, lalu ketikkan `npm install -g @angular/cli`

Setelah Angular berhasil diinstal, ketikkan `ng version` atau `ng v` di CMD untuk mengecek apakah Angular telah terinstall sempurna.

Oke, proses instalasi telah selesai, waktunya membuat project pertama kita.

Pertama, buat sebuah folder di direktori kesukaan anda. Lalu, ketikkan **ng new first-app** di CMD atau terminal. Artinya kita akan membuat project baru bernama "first app".

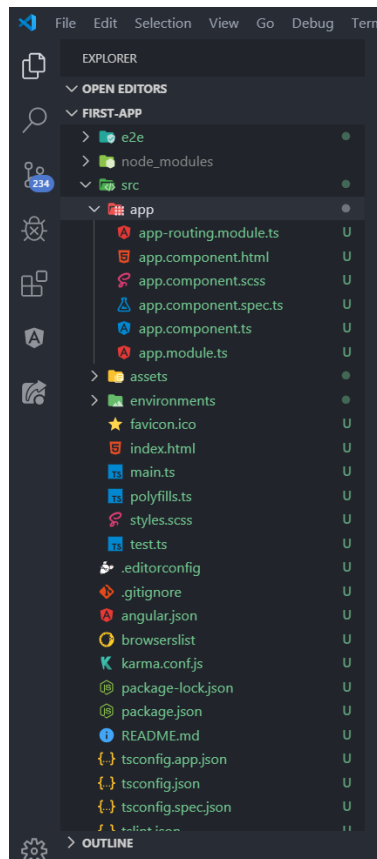
```
F:\SEMESTER 7 TEMP\Keisengan\testAngular>ng new first-app
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use?
  CSS
> SCSS  [ https://sass-lang.com/documentation/syntax#scss ]
  Sass  [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less  [ http://lesscss.org ]
  Stylus [ http://stylus-lang.com ]
```

Dalam hal ini, kita akan menggunakan **SCSS**. SCSS menggunakan *syntax* yang sama dengan CSS. Hanya saja SCSS dinilai lebih fleksibel dan ter-update oleh para *developer*.

Masuklah ke direktori tempat anda membuat project angular dan ketikkan perintah **code** . pada terminal atau CMD untuk membuka *code editor* kesukaan anda.

```
F:\SEMESTER 7 TEMP\Keisengan\testAngular>cd first-app
F:\SEMESTER 7 TEMP\Keisengan\testAngular\first-app>code .
```

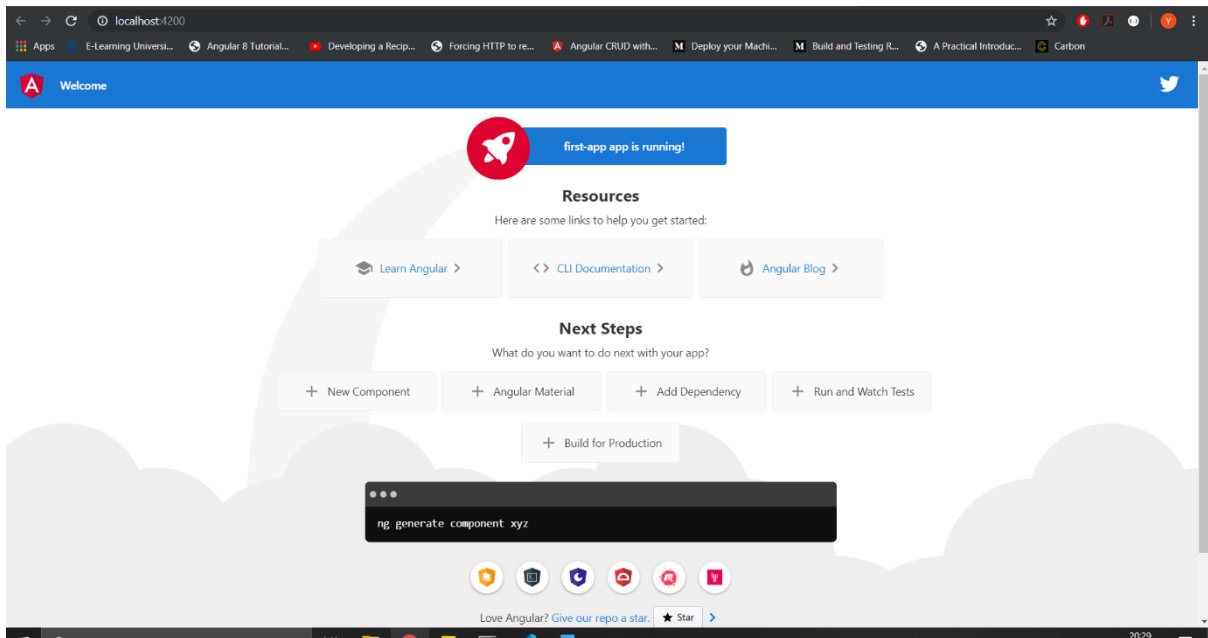
Kira-kira seperti inilah struktur project kita.



Kita akan lebih banyak bermain di folder **app**.

Untuk mengetes project Angular kita, silahkan ketikkan **ng serve -o** di terminal/CMD untuk menjalankan aplikasi web kita.

```
F:\SEMESTER 7 TEMP\Keisengan\testAngular\first-app>ng serve -o
```



Beginilah kira-kira tampilan awal web kita.

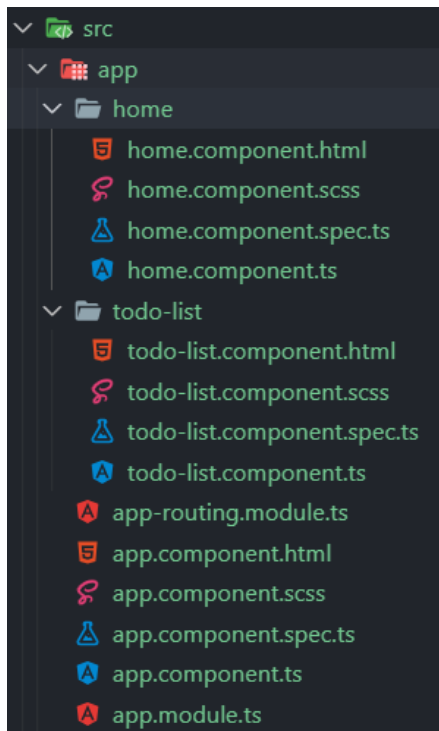
Pada project Angular, kita **tidak perlu refresh** browser ketika kita mengedit kodingan kita, karena ketika kita save projectnya, maka **otomatis akan refresh browser**.

Berikutnya, kita akan membuat halaman baru bernama **todo-list**. Ketikkan **ng generate component todo-list** di CMD. Karena CMD kita masih digunakan untuk ng serve -o, silahkan buka CMD baru dan ketikkan perintah untuk generate halaman todo-list.

```
F:\SEMESTER 7 TEMP\Keisengan\testAngular\first-app>ng generate component todo-list
```

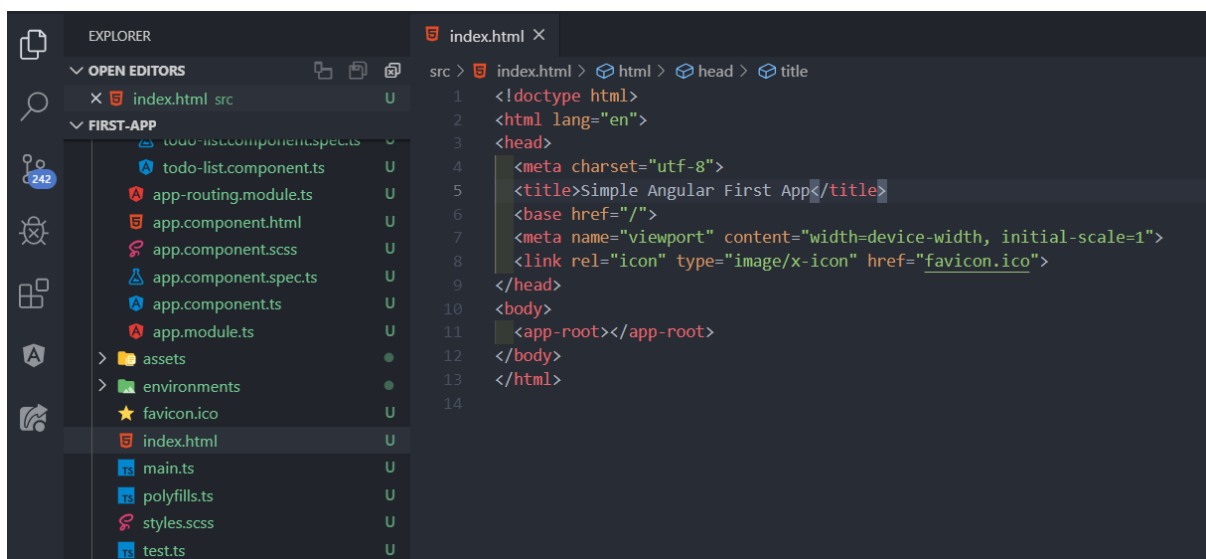
Oh iya, kita lupa membuat halaman Home. Lakukan hal yang sama seperti kalian membuat halaman todo-list.

Seperti inilah stuktur project kita setelah menambah halaman web.

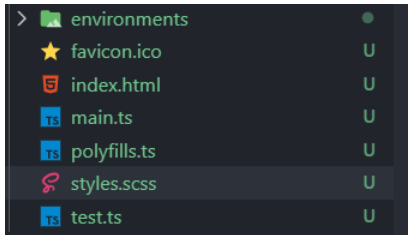


LET'S CODE

Kita masuk ke file **index.html** dan ubahlah kodingan seperti pada gambar.

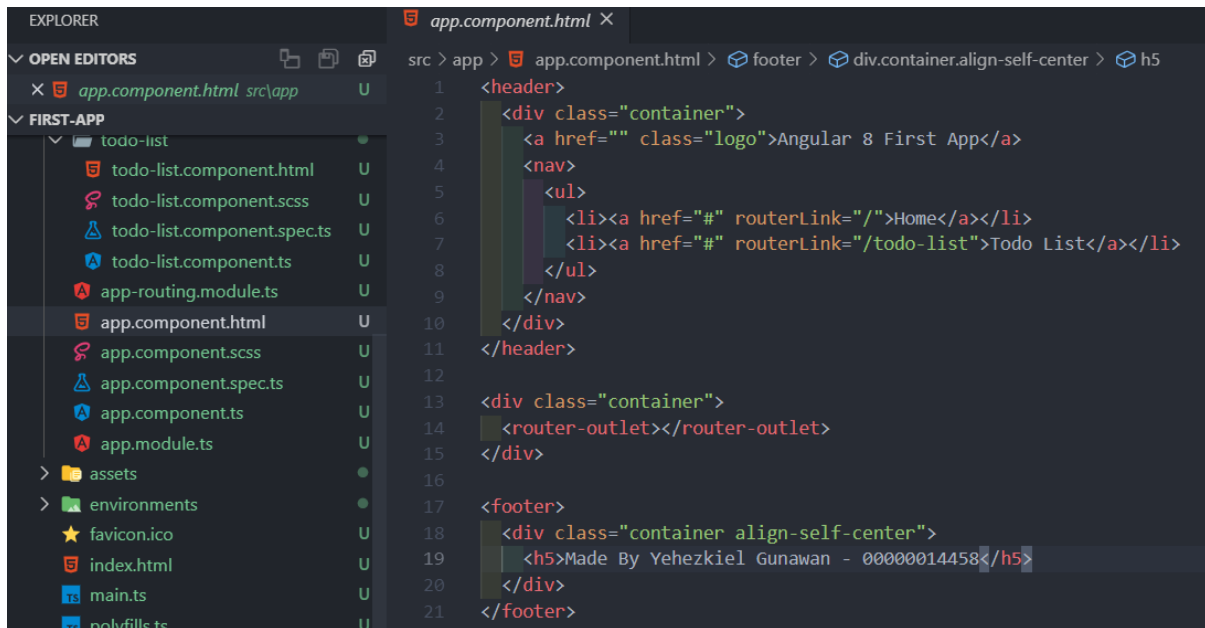


Setelah itu, bukalah file **style.scss** dan ubahlah isi kodingan seperti pada gambar.



```
1 /* You can add global styles to this file, and also import other style files */
2 @import url('https://fonts.googleapis.com/css?family=Nunito:400,700&display=swap');
3
4 $primary: rgb(111, 0, 255);
5
6 body {
7   margin: 0;
8   font-family: 'Nunito', 'sans-serif';
9   font-size: 18px;
10 }
11
12 .container{
13   width: 80%;
14   margin: 0 auto;
15 }
16
17 header{
18   background: $primary;
19   padding: 1em 0;
20
21   a{
22     color: white;
23     text-decoration: none;
24   }
25
26   a.logo{
27     font-weight: bold;
28   }
29
30   nav{
31     float: right;
32
33     ul{
34       list-style-type: none;
35       margin: 0;
36       display: flex;
37
38       li a{
39         padding: 1em;
40
41         &:hover{
42           background: darken($primary, 10%);
43         }
44       }
45     }
46   }
47 }
48
49 h1{
50   margin-top: 2em;
51 }
52
```

Bukalah file **app.component.html** dan ubahlah kodingan seperti pada gambar. **Ubah footer sesuai NIM dan Nama masing-masing.**

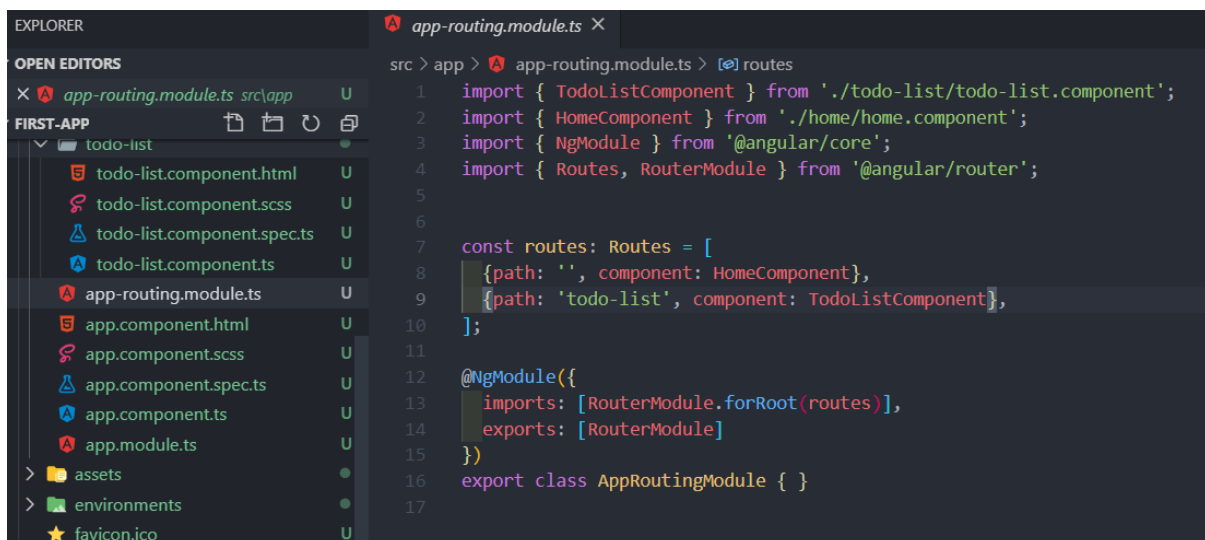


```

1 <header>
2   <div class="container">
3     <a href="" class="logo">Angular 8 First App</a>
4     <nav>
5       <ul>
6         <li><a href="#" routerLink="/">Home</a></li>
7         <li><a href="#" routerLink="/todo-list">Todo List</a></li>
8       </ul>
9     </nav>
10  </div>
11 </header>
12
13 <div class="container">
14   <router-outlet></router-outlet>
15 </div>
16
17 <footer>
18   <div class="container align-self-center">
19     <h5>Made By Yehezkiel Gunawan - 00000014458</h5>
20   </div>
21 </footer>

```

Ubah juga file **app-routing.module.ts** seperti gambar di bawah.



```

1 import { TodoListComponent } from './todo-list/todo-list.component';
2 import { HomeComponent } from './home/home.component';
3 import { NgModule } from '@angular/core';
4 import { Routes, RouterModule } from '@angular/router';
5
6
7 const routes: Routes = [
8   {path: '', component: HomeComponent},
9   {path: 'todo-list', component: TodoListComponent},
10 ];
11
12 @NgModule({
13   imports: [RouterModule.forRoot(routes)],
14   exports: [RouterModule]
15 })
16 export class AppRoutingModule { }
17

```

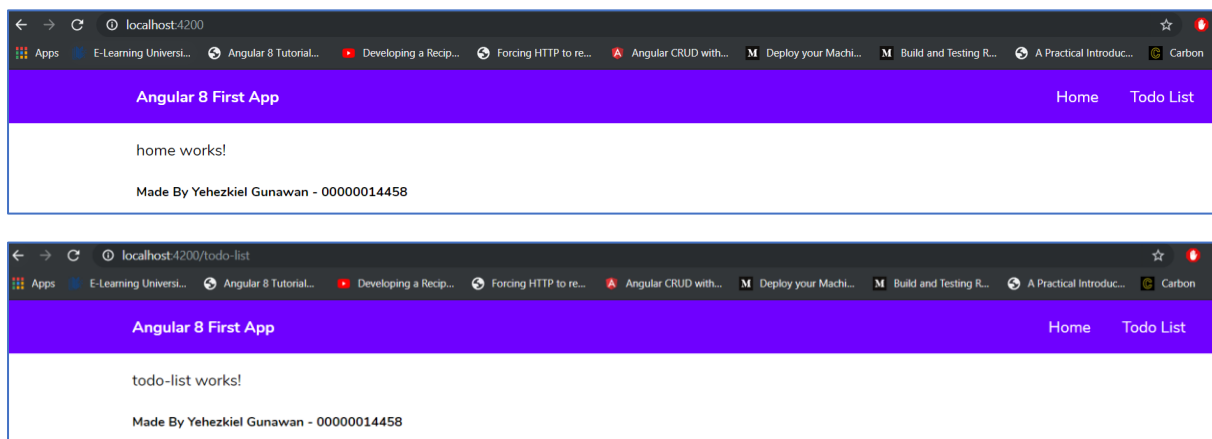
Ubah juga file **app.module.ts** seperti pada gambar.

```

1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { TodoListComponent } from './todo-list/todo-list.component';
7  import { HomeComponent } from './home/home.component';
8  import { FormsModule, ReactiveFormsModule } from '@angular/forms';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     TodoListComponent,
14     HomeComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule,
19     FormsModule,
20     ReactiveFormsModule
21   ],
22   providers: [],
23   bootstrap: [AppComponent]
24 })
25 export class AppModule { }
26

```

Lihatlah hasilnya di browser.



Ok, sekarang kita sudah menyelesaikan “pengaturan” dan *styling* global kita. Di halaman berikutnya kita akan mulai masuk ke halaman home.

Ubahlah file **home.component.html** seperti pada gambar di bawah.

```

1 <h1>Welcome!</h1>
2
3 <div class="play-container">
4   <p>You've clicked <span (click)="countClick()">this</span> {{clickCounter}} </p>
5 </div>
6
7 <div class="play-container">
8   <p>
9     <input type="text" [(ngModel)]="name">
10    <br>
11    <strong>You said: </strong> {{name}}
12  </p>
13 </div>
14
15 <div class="play-container" [ngClass]="setClasses()">
16   <ng-template [ngIf]="clickCounter > 4" [ngIfElse]="none">
17     <p>The click counter <strong>IS GREATER</strong> than 4.</p>
18   </ng-template>
19
20   <ng-template #none>
21     <p>The click counter is <strong>not greater</strong> than 4.</p>
22   </ng-template>
23
24 </div>
25
26

```

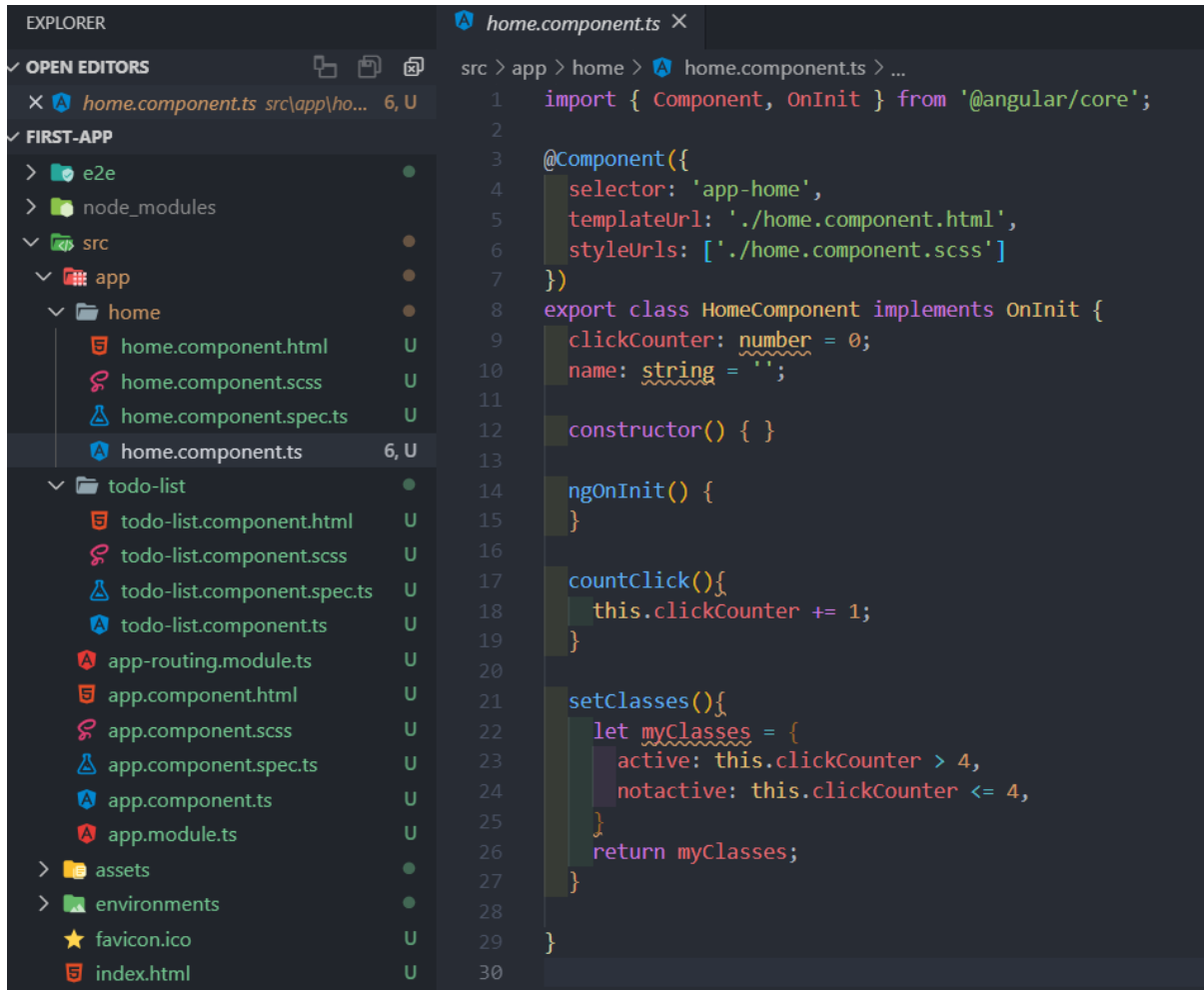
Mari kita *styling* halaman home di **home.component.scss**

```

1 span {
2   font-weight: bold;
3   background: lightgray;
4   padding: .3em, .8em;
5   cursor: pointer;
6   user-select: none;
7 }
8
9 .play-container {
10  padding: 3em;
11  border: 1px solid lightgray;
12  margin-bottom: 1em;
13
14  input {
15    padding: 1em;
16    margin-bottom: 2em;
17  }
18 }
19
20 .active {
21   background-color: yellow;
22   border: 4px solid black;
23 }
24
25 .notactive {
26   background-color: lightgray;
27 }
28

```

Mari kita tambahkan *logic* untuk halaman home di file `home.component.ts` seperti pada gambar di bawah.

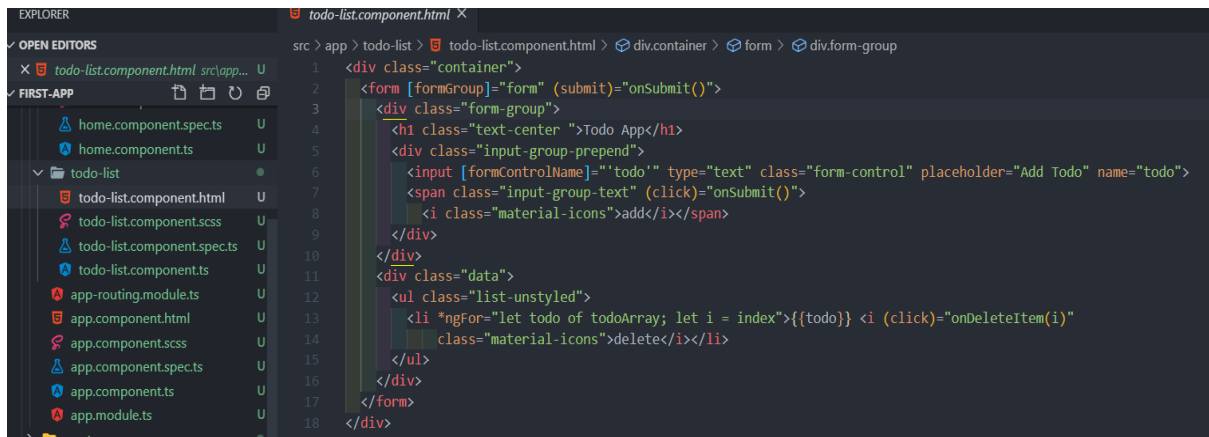


The screenshot shows the Visual Studio Code interface. On the left, the Explorer panel displays the project structure under 'FIRST-APP'. The 'home' folder is expanded, showing files: `home.component.html`, `home.component.scss`, `home.component.spec.ts`, and `home.component.ts`. The `home.component.ts` file is selected. The main editor area shows the content of `home.component.ts` with line numbers 1 through 30. The code defines an Angular component named `HomeComponent` with a selector `'app-home'`, a template URL `./home.component.html`, and style URLs `['./home.component.scss']`. It implements the `OnInit` interface. The component has a `clickCounter` property initialized to 0 and a `name` property initialized to an empty string. The `constructor()` method is empty. The `ngOnInit()` method is also empty. The `countClick()` method increments `this.clickCounter` by 1. The `setClasses()` method defines a `myClasses` object with `active` (true if `clickCounter > 4`) and `notactive` (true if `clickCounter <= 4`).

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-home',
5    templateUrl: './home.component.html',
6    styleUrls: ['./home.component.scss']
7  })
8  export class HomeComponent implements OnInit {
9    clickCounter: number = 0;
10   name: string = '';
11
12   constructor() { }
13
14   ngOnInit() {
15   }
16
17   countClick(){
18     this.clickCounter += 1;
19   }
20
21   setClasses(){
22     let myClasses = {
23       active: this.clickCounter > 4,
24       notactive: this.clickCounter <= 4,
25     }
26     return myClasses;
27   }
28
29 }
30
```

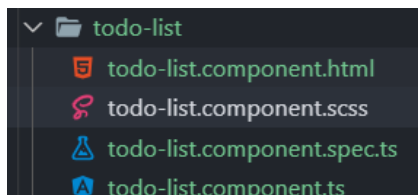
Lihatlah hasilnya di browser dan jalankan. Apabila anda membutuhkan penjelasan segera tanyakan kepada asisten lab.

Sekarang kita akan bermain di halaman to do list. Silahkan ubah **todo-list.component.html** seperti gambar di bawah.



```
1 <div class="container">
2   <form [formGroup]="form" (submit)="onSubmit()">
3     <div class="form-group">
4       <h1 class="text-center">Todo App</h1>
5       <div class="input-group-prepend">
6         <input [formControlName]="todo" type="text" class="form-control" placeholder="Add Todo" name="todo">
7         <span class="input-group-text" (click)="onSubmit()">
8           <i class="material-icons">add</i></span>
9       </div>
10    </div>
11    <div class="data">
12      <ul class="list-unstyled">
13        <li *ngFor="let todo of todoArray; let i = index">{{todo}} <i (click)="onDeleteItem(i)"
14          class="material-icons">delete</i></li>
15      </ul>
16    </div>
17  </form>
18 </div>
```

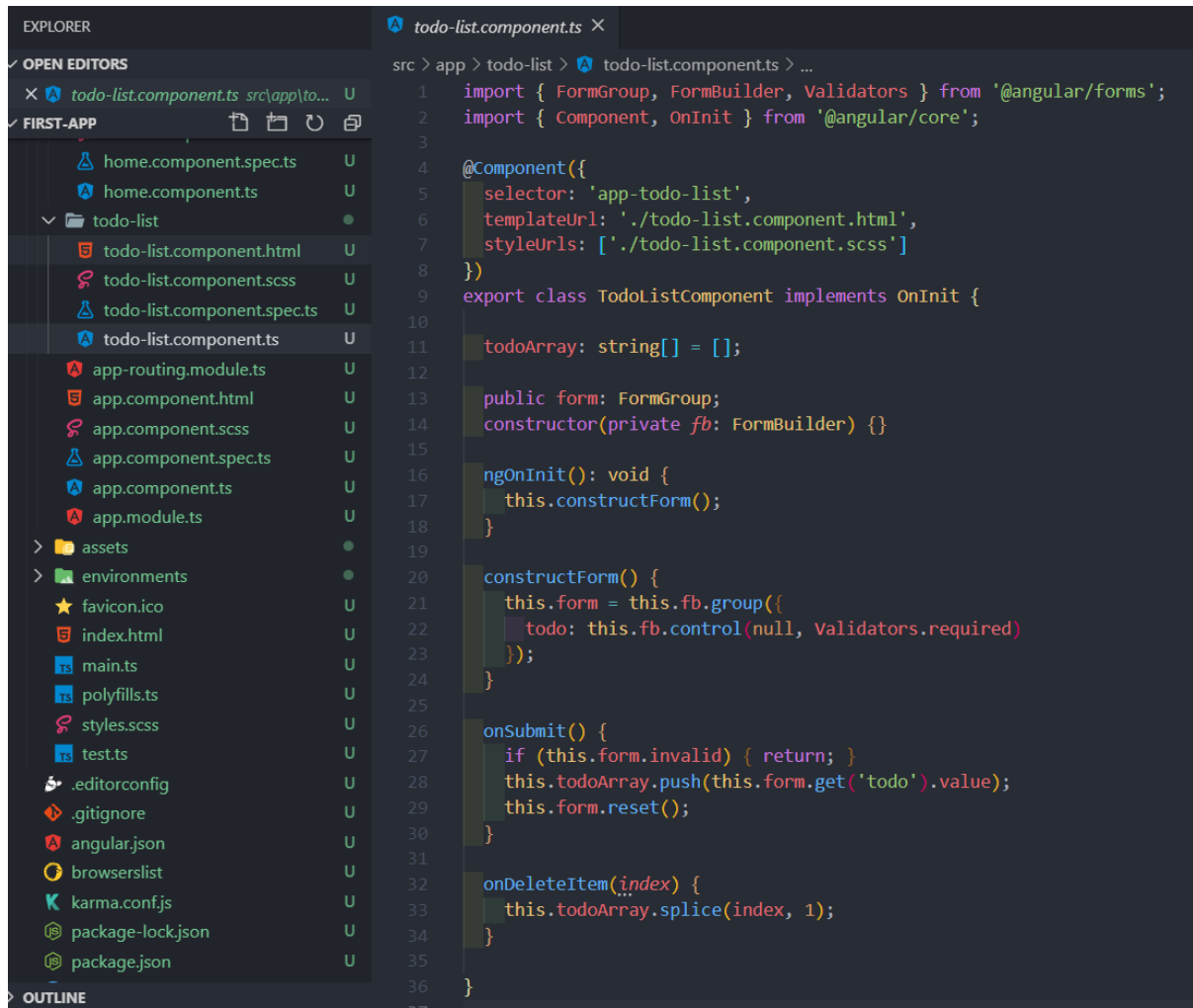
Styling lah file **todo-list.component.scss** seperti gambar di bawah.



BERLANJUT KE HALAMAN BERIKUTNYA

```
1  /*Google fonts*/
2  @import url('https://fonts.googleapis.com/css?family=Raleway');
3
4  body {
5    padding: 0;
6    margin: 0;
7  }
8
9  form {
10   max-width: 30em;
11   margin: 4em auto;
12   position: relative;
13   background: #F4F4F4;
14   padding: 2em 3em;
15   overflow: hidden;
16 }
17
18 form h1 {
19   font-family: "Raleway";
20   color: #F97300;
21 }
22
23 form input[type=text]::placeholder {
24   font-family: "Raleway";
25   color: #666;
26 }
27
28 form .data {
29   margin-top: 1em;
30 }
31
32 form .data li {
33   background: #FFF;
34   border-left: 4px solid #F97300;
35   padding: 1em;
36   margin: 1em auto;
37   color: #666;
38   font-family: "Raleway";
39 }
40
41 form .data li i {
42   float: right;
43   color: #888;
44   cursor: pointer;
45 }
46
47 form .input-group-text {
48   background: #F97300;
49   border-radius: 50%;
50   width: 5em;
51   height: 5em;
52   padding: 1em 23px;
53   color: #fff;
54   position: absolute;
55   right: 13px;
56   top: 68px;
57   cursor: pointer;
58 }
59
60 form .input-group-text i {
61   font-size: 2em;
62 }
63
64 form .form-control {
65   height: 3em;
66   font-family: "Raleway";
67 }
68
69 form .form-control:focus {
70   box-shadow: 0;
71 }
72
```

Ubahkan file **todo-list.component.ts** seperti gambar di bawah.



Lihat hasil project di web browser dan minta penjelasan asisten lab. Karena ini sangat menunjang project UAS anda.

Source :

<https://github.com/brampeirs/todo-app-angular-7>

https://www.youtube.com/watch?v=_TLhUCjY9iA **(Angular 8 Crash Course)**